git-multirepo

# multi init [--extras]

Initialize the current repository as a multirepo project.

**--extras** Only initialize extras (hooks, gitconfig, gitattributes, etc.)

# multi add <path\_to\_sibling\_repo>

Track an additional dependency with multirepo.

# multi branch <branch\_name> [--force] [--no-push]

Create and/or checkout a new branch for all repos.

**--force** Force creating the branch even if the repos contain uncommmitted changes.

**--no-track** Do not configure as a remote-tracking branch on creation.

# multi checkout <ref> [--latest|--exact]

Checks out the specified commit or branch of the main repo and checks out matching versions of all dependencies.

**--latest** Checkout the HEAD of each dependency branch (as recorded in the lock file) instead of the exact required commits.

**--exact** Checkout the exact specified ref for each repo, regardless of what's stored in the lock file.

# multi clone <url> <directory\_name> [<ref>]

Clones the specified repository in a subfolder (<directory\_name>), optionally checks out the specified ref (<ref>) then installs it.

# multi install [--hooks]

Clones and checks out dependencies as defined in the version-controlled multirepo metadata files and installs git-multirepo's local git hooks.

**--hooks** Only install local git hooks.

# multi remove <path\_to\_sibling\_repo> [--force]

Removes the specified dependency from multirepo.

**--force** Delete the dependency on disk in addition to removing it from the multirepo config.

# multi update [--force] [--commit]

Removes the specified dependency from multirepo.

**--force** Update the lock file even if dependencies contain uncommitted changes.

**--commit** Commit the lock file after updating it.

# multi uninit

Removes all traces of multirepo in the current multirepo repository.